Large-Scale Cubic-Scaling Random Phase Approximation Correlation Energy Calculations Using a Gaussian Basis

Wilhelm, Jan ; Seewald, Patrick ; Del Ben, Mauro ; Hutter, Jürg

DOI: https://doi.org/10.1021/acs.jctc.6b00840

Large-scale cubic-scaling RPA correlation energy calculations using a Gaussian basis

Jan Wilhelm,*,† Patrick Seewald,† Mauro Del Ben,‡ and Jürg Hutter*,†

†*Department of Chemistry and National Centre for Computational Design and Discovery of Novel Materials (MARVEL), University of Zurich, 8057 Zurich, Switzerland*

‡*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States*

E-mail: jan.wilhelm@chem.uzh.ch; hutter@chem.uzh.ch

Abstract

We present an algorithm for computing the correlation energy in the random phase approximation (RPA) in a Gaussian basis requiring $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ memory. The method is based on the resolution of the identity (RI) with the overlap metric, a reformulation of RI-RPA in the Gaussian basis, imaginary time and imaginary frequency integration techniques and the use of sparse linear algebra. Additional memory reduction without extra computations can be achieved by an iterative scheme which overcomes the memory bottleneck of canonical RPA implementations. We report a massively parallel implementation which is the key for the application to large systems. Finally, cubic-scaling RPA is applied to a thousand water molecules using a correlation-consistent triple-zeta quality basis.

1 Introduction

The random phase approximation (RPA) for computing electron correlation energies[1,2] has emerged as an accurate tool for predicting properties of isolated molecules[3–11] and condensed phase systems[12–32]. In its simplest form, the RPA total energy is the sum of the Hartree-Fock (HF) energy E^{HF} and the RPA correlation energy $E_{\mathrm{c}}^{\mathrm{RPA}}$,[33]

$$E = E^{\mathrm{HF}}[\{\psi_n\}] + E_{\mathrm{c}}^{\mathrm{RPA}}[\{\psi_n, \varepsilon_n\}]\,. \tag{1}$$

The RPA total energy is typically evaluated after a self-consistent field (SCF) procedure, i.e., the converged molecular orbitals ψ_n from the SCF and their corresponding eigenvalues ε_n are employed to evaluate the HF energy and the RPA correlation energy. RPA combines a number of attractive features, most importantly that long-range dispersion is included – in contrast to semilocal density functionals.[33–40]

The drawback connected with RPA is the computational cost: For canonical implementations of RPA in a plane-waves basis[41–43] or in a localized basis within the resolution of the identity (RI)[33,44–46], the computational cost of RPA scales as $\mathcal{O}(N^4)$ with respect to the system size N. Recently, low-scaling RPA algorithms have been explored: Moussa[47] employed the connection of RPA to coupled-cluster theory for constructing an $\mathcal{O}(N^3)$ scaling RPA+SOSEX algorithm. In this case, the cubic scaling has been demonstrated employing chains of hydrogen atoms. Kaltak *et al.*[48] use a plane-wave basis, minimax grids in imaginary frequency and imaginary time to arrive at a cubic-scaling algorithm. They applied their implementation for studying supercells of silicon[49] containing up to 256 atoms. Linear-scaling RPA algorithms have also been reported[50–53] which either rely on localization techniques[50], stochastic sampling[51,52] or sparsity of density matrices[53]. Large-scale applications of linear-scaling RPA have only been reported so far in Ref. 51 and 52 using stochastic sampling.

In this work, we employ localized Gaussian basis functions. We combine the $\mathcal{O}(N^4)$ RI-RPA method by Eshuis *et al.*[33] and the minimax grids in imaginary frequency and imaginary

time from Kaltak *et al.*[48] To achieve the cubic scaling algorithm, we rely on the sparsity introduced by the RI approximation with the overlap metric[54–56] combined with the back transformation from occupied and virtual orbitals to Gaussian atomic orbitals[53]. Practically, the sparsity is efficiently handled by the DBCSR (Distributed Block Compressed Sparse Row) library[57,58]. Stochastic sampling, sparsity of density matrices and localization techniques are not necessary for the $\mathcal{O}(N^3)$ scaling behavior in our algorithm.

The manuscript is organized as follows: In Sec. 2, we review the resolution of the identity with the overlap metric. As we explain in Sec. 3, cubic-scaling RPA is well-known in a formulation in imaginary time and in real space. We use this formulation of cubic-scaling RPA to construct a cubic-scaling algorithm using Gaussian basis functions (Sec. 4), while we focus on the parallel implementation in Sec. 5. Benchmark calculations on the accuracy and the scaling of the $\mathcal{O}(N^3)$ RPA method are reported in Secs. 6 and 7.

2 Resolution-of-the-identity approximation (RI) using the overlap metric

The following index notation has been adopted: i, j (a, b) refer to occupied (virtual) molecular orbitals (MOs) ψ; $\mu, \nu, \lambda, \sigma$ to primary Gaussian basis functions ϕ and P, Q to auxiliary Gaussian RI basis functions φ. The number of primary Gaussian basis functions is referenced as N_{prim}, the number of RI basis functions as N_{RI} and the system size as abstract symbol N, where N_{prim} and N_{RI} are both proportional to N. The spin index has been dropped for convenience.

Four-center electron repulsion integrals (4c-ERIs) are of central importance for computing the RPA correlation energy. These integrals, in Mulliken notation, are defined as

$$(\mu\nu|\lambda\sigma) := \int_{\Omega} d\mathbf{r} \int_{\mathbb{R}^3} d\mathbf{r}' \phi_\mu^{\text{P}}(\mathbf{r}')\phi_\nu^{\text{P}}(\mathbf{r}')\phi_\lambda^{\text{P}}(\mathbf{r})\phi_\sigma^{\text{P}}(\mathbf{r})v(\mathbf{r},\mathbf{r}') \tag{2}$$

where $v(\mathbf{r},\mathbf{r}')=1/|\mathbf{r}-\mathbf{r}'|$ is the bare Coulomb interaction, Ω the simulation cell and the superscript P indicates that the basis functions are periodically repeated for condensed phase systems,

$$\phi_\mu^{\mathrm{P}}(\mathbf{r}) = \sum_i \phi_\mu(\mathbf{r}-\mathbf{R}_i)\,, \tag{3}$$

where $\mathbf{R}_i$ are the lattice vectors and ϕ_μ is a Gaussian basis function being localized on a single atom. Within the RI approximation based on the overlap metric, the 4c-ERIs are factorized to[54]

$$(\mu\nu|\lambda\sigma)_{\mathrm{RI}} = \sum_{PQRS} (\mu\nu P) S^{-1}_{PQ} V_{QR} S^{-1}_{RS} (S\lambda\sigma)\,. \tag{4}$$

Here, $\mathbf{S}$ denotes the overlap matrix in the RI basis,

$$S_{PQ} = \int_{\mathbb{R}^3} d\mathbf{r}\; \varphi_P^{\mathrm{P}}(\mathbf{r})\varphi_Q(\mathbf{r})\,, \tag{5}$$

and $\mathbf{V}$ the Coulomb matrix in the RI basis,

$$V_{PQ} = \int_\Omega d\mathbf{r} \int_{\mathbb{R}^3} d\mathbf{r}'\varphi_P^{\mathrm{P}}(\mathbf{r})\varphi_Q^{\mathrm{P}}(\mathbf{r}')v(\mathbf{r},\mathbf{r}')\,. \tag{6}$$

The three-center overlap integrals $(\mu\nu P)$ are given by

$$(\mu\nu P) = \int_{\mathbb{R}^3} d\mathbf{r}\, \phi_\mu^{\mathrm{P}}(\mathbf{r})\phi_\nu^{\mathrm{P}}(\mathbf{r})\varphi_P(\mathbf{r})\,. \tag{7}$$

We compute the two- and three-center overlap integrals from Eqs. (5) and (7) employing the Obara-Saika recurrence scheme[59]. In the periodic case, neighboring cells have to be considered for the Obara-Saika scheme as long as the overlap of Gaussians from the unit cell and the replica are non-vanishing.

The Coulomb integrals V_{PQ} are calculated by an integral scheme that has recently been

developed in our group. This method is an analytical variant of the Gaussian and plane waves (GPW) method[44] for periodic ERIs over Gaussian-type functions. We choose a basis of Hermite Gaussians[60] $H_{\mathbf{l},a,\mathbf{A}}(\mathbf{r}) = \partial^{l_x+l_y+l_z}/(\partial A_x^{l_x}\partial A_y^{l_y}\partial A_z^{l_z}) \exp\left(-a(\mathbf{r}-\mathbf{A})^2\right)$ and use the reciprocal space representation of Eq. (6),

$$V_{H_{\mathbf{l},a,\mathbf{A}},H_{\mathbf{m},b,\mathbf{B}}} = \frac{4\pi}{V} \sum_{\mathbf{G}\neq\mathbf{0}} \frac{\widehat{H}_{\mathbf{l},a,\mathbf{A}}(\mathbf{G})\widehat{H}_{\mathbf{m},b,\mathbf{B}}(-\mathbf{G})}{|\mathbf{G}|^2} \,. \tag{8}$$

The Fourier transforms $\widehat{H}(\mathbf{G})$ are conveniently given by Cartesian Gaussians. We expand the reciprocal space Coulomb potential into Gaussians by the minimax approximation[61]

$$\frac{1}{|\mathbf{G}|^2} \approx \sum_{i=1}^{n} \omega_i \exp(-\alpha_i |\mathbf{G}|^2) \qquad \text{for } |\mathbf{G}| \in [G_{\min}, G_{\mathrm{c}}] \,, \tag{9}$$

where $G_{\min}$ is the minimum of $|\mathbf{G}| \neq 0$ and G_{c} is the plane wave cutoff. For an orthorhombic cell, the resulting lattice sum factorizes in the Cartesian directions such that it reduces to a product of three one-dimensional sums. In a similar fashion as in Ewald summation[62], the reciprocal lattice sum can alternatively be carried out over direct lattice vectors $\mathbf{R}$ by Poisson summation. This choice of summing over direct or reciprocal lattice vectors ensures rapid convergence for arbitrary Gaussian exponents.

As in Ref. 44, we neglect the $\mathbf{G}=\mathbf{0}$ component of the summation in Eq. (8) which is justified by the following argument: In RPA, only 4c-ERIs of the form $(ia|jb)$ with $\psi_{ia}(\mathbf{G}=\mathbf{0})=0$ $[\psi_{ia}(\mathbf{r})=\psi_i(\mathbf{r})\psi_a(\mathbf{r})]$ are appearing. Since the RI basis is supposed to span the space $\{\psi_{ia}\}$ of functions with vanishing $\mathbf{G}=\mathbf{0}$ component, the RI basis functions can be chosen with vanishing $\mathbf{G}=\mathbf{0}$ component.

Since the Coulomb matrix $\mathbf{V}$ is positive definite, its Cholesky decomposition can be computed as

$$V_{PQ} = \sum_{R} L_{PR} L_{RQ}^{\mathrm{T}} \,. \tag{10}$$

In this way, the RI factorization from Eq. (4) can be expressed in a compact form as

$$(\mu\nu|\lambda\sigma)_{\mathrm{RI}} = \sum_{P} B_P^{\mu\nu} B_P^{\lambda\sigma} \,, \tag{11}$$

where $\mathbf{B}$ is given by

$$B_P^{\mu\nu} = \sum_{QR} (\mu\nu Q) S_{QR}^{-1} L_{RP} \,. \tag{12}$$

We define the $N_{\mathrm{RI}} \times N_{\mathrm{RI}}$ matrix $\mathbf{K}$,

$$\mathbf{K} = \mathbf{S}^{-1}\mathbf{L} \,, \tag{13}$$

and Eq. (12) simplifies to

$$B_P^{\mu\nu} = \sum_{Q} (\mu\nu Q) K_{QP} \,. \tag{14}$$

If required, we transform $B_P^{\mu\nu}$ from pairs $\mu\nu$ of Gaussian basis functions to occupied-virtual pairs ia employing the MO coefficients $C_{\mu n}$:

$$B_P^{ia} = \sum_{\mu\nu} C_{\mu i} C_{\nu a} B_P^{\mu\nu} \,. \tag{15}$$

3 Cubic-scaling RPA with real-space density response

The correlation energy in the random phase approximation can be computed as[2,33,48,63]

$$E_{\mathrm{c}}^{\mathrm{RPA}} = \frac{1}{2\pi} \int_0^\infty d\omega \; \mathrm{Tr}\left[\ln\left(1 - \chi(i\omega)v\right) + \chi(i\omega)v\right] \tag{16}$$

where $v(\mathbf{r},\mathbf{r}') = 1/|\mathbf{r}-\mathbf{r}'|$ is the bare Coulomb interaction and $\chi(\mathbf{r},\mathbf{r}',i\omega)$ the density response in imaginary frequency:

$$\chi(\mathbf{r},\mathbf{r}',i\omega) = 2\sum_{i}^{\text{occ}}\sum_{a}^{\text{virt}} \psi_a(\mathbf{r}')\psi_i(\mathbf{r}')\psi_i(\mathbf{r})\psi_a(\mathbf{r})\,\frac{\varepsilon_i-\varepsilon_a}{\omega^2+(\varepsilon_i-\varepsilon_a)^2}\,. \tag{17}$$

The drawback of employing Eq. (17) to compute the density response function is the $\mathcal{O}(N^4)$ computational cost (N: system size) since the number of occupied and virtual orbitals and the space coordinates $\mathbf{r}$ and $\mathbf{r}'$ are each growing linearly with N. In contrast, the computation of the density response in imaginary time,[48,64]

$$\chi(\mathbf{r},\mathbf{r}',i\tau) = \sum_{i}^{\text{occ}} \psi_i(\mathbf{r}')\psi_i(\mathbf{r})e^{-|(\varepsilon_i-\varepsilon_\text{F})\tau|}\sum_{a}^{\text{virt}} \psi_a(\mathbf{r}')\psi_a(\mathbf{r})e^{-|(\varepsilon_a-\varepsilon_\text{F})\tau|}\,, \tag{18}$$

only requires $\mathcal{O}(N^3)$ operations since the summation over occupied and virtual orbitals are decoupled and can be executed separately. ε_F in Eq. (18) refers to the Fermi energy.

The density response is symmetric in time and frequency, $\chi(\mathbf{r},\mathbf{r}',i\omega) = \chi(\mathbf{r},\mathbf{r}',-i\omega)$ and $\chi(\mathbf{r},\mathbf{r}',i\tau) = \chi(\mathbf{r},\mathbf{r}',-i\tau)$. As a consequence, the Fourier transforms from imaginary frequency to imaginary time and vice versa simplify to a cosine transformation[48]:

$$\chi(\mathbf{r},\mathbf{r}',i\omega) = 2\int_0^\infty d\tau\,\chi(\mathbf{r},\mathbf{r}',i\tau)\cos(\tau\omega)\,, \tag{19}$$

$$\chi(\mathbf{r},\mathbf{r}',i\tau) = \frac{1}{\pi}\int_0^\infty d\omega\,\chi(\mathbf{r},\mathbf{r}',i\omega)\cos(\tau\omega)\,. \tag{20}$$

Despite the simplicity of this formulation, the size of the real space coordinate $\mathbf{r}$ is of the same order as the size of a plane-wave basis[48] and thus can easily exceed millions of elements even for a relatively small cell. One of the reasons of reformulating these equations in a Gaussian basis is to reduce the size of the density response matrix without significant loss of accuracy.

4 Cubic-scaling RPA in a Gaussian basis

4.1 Quartic-scaling RPA in a Gaussian basis

Eshuis *et al.*[33] applied the RI [Eq. (11)] to Eq. (16) and obtained

$$E_{\mathrm{c}}^{\mathrm{RPA}} = \frac{1}{2\pi} \int_0^\infty d\omega \, \mathrm{Tr} \left[\ln(\mathbf{1} + \mathbf{Q}(\omega)) - \mathbf{Q}(\omega) \right] \tag{21}$$

where $\mathbf{Q}(\omega)$ is a matrix of size $N_{\mathrm{RI}} \times N_{\mathrm{RI}}$ and is given by

$$Q_{PQ}(\omega) = 2 \sum_i^{\mathrm{occ}} \sum_a^{\mathrm{virt}} B_P^{ia} \frac{\varepsilon_a - \varepsilon_i}{\omega^2 + (\varepsilon_a - \varepsilon_i)^2} B_Q^{ia} \,, \tag{22}$$

where B_P^{ia} is defined in Eq. (15). For a derivation of Eq. (21) from Eq. (16), we refer to the supporting information. In the $\mathcal{O}(N^3)$ implementation, we do not compute $\mathbf{Q}(\omega)$ by means of Eq. (22) due to the $\mathcal{O}(N^4)$ computational cost. Instead, we compute $\mathbf{Q}(\tau)$ as presented in the following and we obtain $\mathbf{Q}(\omega)$ by the cosine transform adapted from Eq. (19):

$$Q_{PQ}(\omega) = 2 \int_0^\infty d\tau \, Q_{PQ}(\tau) \cos(\tau\omega) \,. \tag{23}$$

We obtain a representation for $\mathbf{Q}(\tau)$ by comparing Eqs. (17), (18) and (22):

$$Q_{PQ}(\tau) = \sum_i^{\mathrm{occ}} \sum_a^{\mathrm{virt}} B_P^{ia} \, e^{-|(\varepsilon_i - \varepsilon_{\mathrm{F}})\tau|} e^{-|(\varepsilon_a - \varepsilon_{\mathrm{F}})\tau|} \, B_Q^{ia} \,. \tag{24}$$

The computation of $\mathbf{Q}(\tau)$ according to Eq. (24) still scales as $\mathcal{O}(N^4)$.

4.2 Cubic-scaling reformulation of RPA in the Gaussian basis

To arrive at a $\mathcal{O}(N^3)$ scaling algorithm, we transform B_P^{ia} in Eq. (24) from occupied-virtual pairs ia to pairs $\mu\nu$ of primary basis functions:

$$B_P^{ia} = \sum_{\mu\nu} B_P^{\mu\nu} C_{\mu i} C_{\nu a} \,, \tag{25}$$

where we used the MO coefficients $C_{\mu n}$ being defined as $\psi_n(\mathbf{r}) = \sum_\mu C_{\mu n} \phi_\mu(\mathbf{r})$. By inserting Eq. (25) into Eq. (24), we separate the summation over occupied and virtual states which is the key for the $\mathcal{O}(N^3)$ implementation:

$$Q_{PQ}(\tau) = \sum_{\mu\nu\lambda\sigma} B_P^{\mu\nu} B_Q^{\lambda\sigma} \sum_i^{\text{occ}} C_{\mu i} C_{\lambda i} e^{-|(\varepsilon_i - \varepsilon_\text{F})\tau|} \sum_a^{\text{virt}} C_{\nu a} C_{\sigma a} e^{-|(\varepsilon_a - \varepsilon_\text{F})\tau|} \,. \tag{26}$$

We introduce the pseudo-density matrices[65–67] $D_{\mu\lambda}^{\text{occ}}(\tau)$ and $D_{\nu\sigma}^{\text{virt}}(\tau)$ which are computed in $\mathcal{O}(N^3)$ operations as

$$\begin{aligned} D_{\mu\lambda}^{\text{occ}}(\tau) &= \sum_i^{\text{occ}} C_{\mu i} C_{\lambda i} e^{-|(\varepsilon_i - \varepsilon_\text{F})\tau|} \,, \\ D_{\nu\sigma}^{\text{virt}}(\tau) &= \sum_a^{\text{virt}} C_{\nu a} C_{\sigma a} e^{-|(\varepsilon_a - \varepsilon_\text{F})\tau|} \,. \end{aligned} \tag{27}$$

Inserting the definitions of $D_{\mu\lambda}^{\text{occ}}$ and $D_{\nu\sigma}^{\text{virt}}$ [Eq. (27)] and the definition of $B_P^{\mu\nu}$ from Eq. (14) into Eq. (26), we obtain

$$Q_{PQ}(\tau) = \sum_R K_{RP} \sum_T K_{TQ} \sum_{\mu\sigma} \left[\sum_\lambda (\lambda\sigma R) D_{\mu\lambda}^{\text{occ}}(\tau) \right] \left[\sum_\nu (\mu\nu T) D_{\nu\sigma}^{\text{virt}}(\tau) \right] \,. \tag{28}$$

We introduce the three-index tensors $M^{\mathrm{occ}}_{\mu\sigma R}(\tau)$ and $M^{\mathrm{virt}}_{\mu\sigma T}(\tau)$:

$$\begin{aligned} M^{\mathrm{occ}}_{\mu\sigma R}(\tau) &= \sum_{\lambda} (\lambda\sigma R) D^{\mathrm{occ}}_{\mu\lambda}(\tau)\,, \\ M^{\mathrm{virt}}_{\mu\sigma T}(\tau) &= \sum_{\nu} (\mu\nu T) D^{\mathrm{virt}}_{\nu\sigma}(\tau)\,. \end{aligned} \tag{29}$$

Both tensors $M^{\mathrm{occ}}_{\mu\sigma R}$ and $M^{\mathrm{virt}}_{\mu\sigma T}$ are computed from Eq. (29) in $\mathcal{O}(N^2)$ operations and in $\mathcal{O}(N)$ operations once the scaled density matrices from Eq. (27) are getting sparse[67] for non-metallic systems being very large in at least one dimension. The reason for the low-scaling behavior $\mathcal{O}(N^2)/\mathcal{O}(N)$ is that the three-center overlap $(\mu\nu T)$ of localized basis functions μ, ν and T is sparse in μ/ν, μ/T and ν/T.[55,56] Inserting the definitions of $M^{\mathrm{occ}}_{\mu\sigma R}$ and $M^{\mathrm{virt}}_{\mu\sigma T}$ in Eq. (28) yields

$$Q_{PQ}(\tau) = \sum_{R} K_{RP} \sum_{T} K_{TQ} \sum_{\mu\sigma} M^{\mathrm{occ}}_{\mu\sigma R}(\tau) M^{\mathrm{virt}}_{\mu\sigma T}(\tau)\,. \tag{30}$$

We introduce the $N_{\mathrm{RI}} \times N_{\mathrm{RI}}$ matrix $\mathbf{P}(\tau)$ with elements

$$P_{RT}(\tau) = \sum_{\mu\sigma} M^{\mathrm{occ}}_{\mu\sigma R}(\tau) M^{\mathrm{virt}}_{\mu\sigma T}(\tau)\,. \tag{31}$$

According to the definition of $M^{\mathrm{occ}}_{\mu\sigma R}$ and $M^{\mathrm{virt}}_{\mu\sigma T}$ in Eq. (29), $M^{\mathrm{occ}}_{\mu\sigma R}$ is sparse in σ/R and $M^{\mathrm{virt}}_{\mu\sigma T}$ is sparse in μ/T and we conclude that the computation of $\mathbf{P}(\tau)$ in Eq. (31) only requires $\mathcal{O}(N^2)$ operations and, once the scaled density matrices from Eq. (27) are getting sparse, only $\mathcal{O}(N)$ operations . By inserting Eq. (31) into Eq. (30), we finally obtain the working expression

$$\mathbf{Q}(\tau) = \mathbf{K}^{\mathrm{T}} \mathbf{P}(\tau) \mathbf{K}\,. \tag{32}$$

The computational complexity of Eq. (32) is growing cubically, $\mathcal{O}(N^3)$. Moreover, all previous steps, Eq. (27), (29) and (31), are at most of $\mathcal{O}(N^3)$ computational cost or in case sparse matrix-matrix multiplication is employed [Eq. (29) and (31)], of $\mathcal{O}(N^2)$ computational cost.

4.3 Minimax time and frequency grid

For the cosine transform of $\mathbf{Q}$ from τ to ω in Eq. (23) and for the subsequent frequency integration to obtain the RPA correlation energy according to Eq. (21), we rely on a time and a frequency grid: In a first step, we compute $Q_{PQ}(\tau_j)$ for a time grid $\{\tau_j\}_{j=1}^{M}$ according to Eq. (32), where M is the number of grid points which is independent of the system size. Then, $Q_{PQ}(\omega_k)$ is obtained for a frequency set $\{\omega_k\}_{k=1}^{M}$ by the cosine transform according to Eq. (23):

$$Q_{PQ}(\omega_k) = 2\sum_{j=1}^{M} \lambda_{kj} Q_{PQ}(\tau_j) \cos(\tau_j \omega_k)\,, \tag{33}$$

where λ_{kj} are the integration weights. The RPA correlation energy is computed by numerically integrating Eq. (21):

$$E_{\mathrm{c}}^{\mathrm{RPA}} = \frac{1}{2\pi}\sum_{k=1}^{M} \sigma_k \left[\ln[\det(\mathbf{1}+\mathbf{Q}(\omega_k))] - \mathrm{Tr}(\mathbf{Q}(\omega_k))\right]\,, \tag{34}$$

where the identity $\mathrm{Tr}[\ln(\mathbf{A})] = \ln[\det(\mathbf{A})]$ has been used which holds for any positive-definite matrix $\mathbf{A}$.[45] We follow the work of Kaltak *et al.*[48] and employ minimax quadratures[68,69] to reduce the number of integration nodes M to 10 – 20. Practically, we employ the pretabulated minimax parameters $\{\omega_k\}$, $\{\sigma_k\}$ and $\{\tau_j\}$ from Ref. 45 which have been created for imaginary-frequency RPA calculations and for Laplace scaled-opposite-spin second-order Møller-Plesset perturbation theory. The integration weights λ_{kj} are computed by a least-square optimization using singular value decomposition[48].

The outline of the cubic-scaling RPA correlation energy algorithm is summarized in Fig. 1.

Compute $S_{PQ} = \int d\mathbf{r}\, \varphi_P(\mathbf{r})\varphi_Q(\mathbf{r})$ $O(N^1)$

Compute $V_{PQ} = \int d\mathbf{r}\, d\mathbf{r}'\, \varphi_P(\mathbf{r})\varphi_Q(\mathbf{r}')|\mathbf{r}-\mathbf{r}'|^{-1}$ $O(N^2)$

Factorize V_{PQ} by Cholesky decomposition: $V_{PQ} = \sum_R L_{PR} L_{RQ}^{\mathrm{T}}$ $O(N^3)$

Invert S_{PQ} with use of Cholesky decomposition $O(N^3)$

Obtain K_{PQ} by matrix-matrix multiplication: $K_{PQ} = \sum_R S_{PR}^{-1} L_{QR}$ $O(N^3)$

Compute $(\mu\nu P) = \int d\mathbf{r}\, \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\varphi_P(\mathbf{r})$ $O(N^1)$

Set up the minimax grids $\{\omega_k\}_{k=1}^M$, $\{\tau_j\}_{j=1}^M$, $\{\lambda_{kj}\}_{k,j=1}^M$ and $\{\sigma_k\}_{k=1}^M$

Get input: MO coefficients $C_{\mu n}$ and eigenvalues ε_n of MOs from SCF

Set $Q_{PQ}(\omega_k) = 0$ for all $\omega_k = \omega_1, \omega_2, \ldots, \omega_M$

Do $\tau_j = \tau_1, \tau_2, \ldots, \tau_M$

$D_{\mu\lambda}^{\mathrm{occ}}(\tau_j) = \sum_i^{\mathrm{occ}} C_{\mu i} C_{\lambda i} e^{-|(\varepsilon_i - \varepsilon_\mathrm{F})\tau_j|}$ $O(MN^3)$

$D_{\nu\sigma}^{\mathrm{virt}}(\tau_j) = \sum_a^{\mathrm{virt}} C_{\nu a} C_{\sigma a} e^{-|(\varepsilon_a - \varepsilon_\mathrm{F})\tau_j|}$ $O(MN^3)$

$M_{\mu\sigma R}^{\mathrm{occ}}(\tau_j) = \sum_\lambda (\lambda\sigma R) D_{\mu\lambda}^{\mathrm{occ}}(\tau_j)$ $O(MN^2)$

$M_{\mu\sigma T}^{\mathrm{virt}}(\tau_j) = \sum_\nu (\mu\nu T) D_{\nu\sigma}^{\mathrm{virt}}(\tau_j)$ $O(MN^2)$

$P_{RT}(\tau_j) = \sum_{\mu\sigma} M_{\mu\sigma R}^{\mathrm{occ}}(\tau_j) M_{\mu\sigma T}^{\mathrm{virt}}(\tau_j)$ $O(MN^2)$

$Q_{PQ}(\tau_j) = \sum_R K_{RP} \sum_T K_{TQ} P_{RT}(\tau_j)$ $O(MN^3)$

Do $\omega_k = \omega_1, \omega_2, \ldots, \omega_M$

Fourier transform from τ to ω:

$Q_{PQ}(\omega_k) = Q_{PQ}(\omega_k) + 2\lambda_{kj} Q_{PQ}(\tau_j)\cos(\tau_j\omega_k)$ $O(M^2N^2)$

End do ω_k

End do τ_j

Set $E_\mathrm{c}^{\mathrm{RPA}} = 0$

Do $\omega_k = \omega_1, \omega_2, \ldots, \omega_M$

Update $E_\mathrm{c}^{\mathrm{RPA}} = E_\mathrm{c}^{\mathrm{RPA}} + \sigma_k \left[\ln[\det(\mathbf{1} + \mathbf{Q}(\omega_k))] - \mathrm{Tr}(\mathbf{Q}(\omega_k))\right]$ $O(MN^3)$

End do ω_k

Figure 1: Pseudocode and associated computational cost (N: system size, M: number of time and frequency points, respectively) of the algorithm to compute the RPA correlation energy in $\mathcal{O}(N^3)$ operations. $\mu, \nu, \lambda, \sigma$ refer to primary Gaussian basis functions, P, Q, R, T to auxiliary Gaussian basis functions, a to virtual molecular orbitals (MOs), i to occupied and n to general ones.

4.4 Comparison to similar work

Overall, the cubic-scaling RPA algorithm presented here is based on similar ideas as the effective-linear-scaling RPA algorithm reported by Schurkus and Ochsenfeld[53]: As in our algorithm, the basis is transformed from occupied and virtual MOs to Gaussian basis functions. Moreover, RI with the overlap metric and an imaginary frequency integration is used. In Ref. 53, a contracted double-Laplace expansion is employed, where we use the Fourier transform from imaginary time to imaginary frequency. In contrast to the double-Laplace expansion, the Fourier transform does not suffer from a divergence for $\omega \to 0$ and $\omega \to \infty$. Moreover, we use minimax grids with a typical number of 15 grid points needed for convergence while in Ref. 53, a Clenshaw-Curtis grid is used with typically 60 grid points. The linear scaling behaviour reported in Ref. 53 relies on the sparsity of the scaled density matrices D^{occ} and D^{virt}. We only exploit this sparsity in Eq. (29), but not in Eq. (27) since the computation time spent for Eq. (27) is far below the time spent for Eq. (32). The reason is that the RI basis is typically three times larger than the primary basis and the matrix $\mathbf{K}$ defined in

Eq. (13) is a dense matrix and therefore, Eq. (32) scales as $N_{\mathrm{RI}}^3 \approx 27 N_{\mathrm{prim}}^3$ where N_{prim} is the number of primary basis functions and Eq. (27) scales as N_{prim}^3.

5 Parallel implementation

Compute $\mathbf{S}^{-1}$ and Cholesky decomposition $\mathbf{L}$ of $\mathbf{V}$ (ScaLAPACK)

Compute and store $\mathbf{K} = \mathbf{S}^{-1}\mathbf{L}$ (ScaLAPACK)

Assign each process a group p

Create ranges $[P_{\mathrm{start}}^p, P_{\mathrm{end}}^p]$ for RI basis functions

Allocate single DBCSR matrix $\mathbf{S}^p$ in the p group (elements $S_{\mu(\sigma P)}^p$, row index μ, combined column index σP for $P \in [P_{\mathrm{start}}^p, P_{\mathrm{end}}^p]$ and all μ, σ; blocks as overlap matrix $\mathbf{S}$, column block sizes scaled by $P_{\mathrm{end}}^p - P_{\mathrm{start}}^p + 1$)

Compute and store $S_{\mu(\sigma P)}^p = (\mu\sigma P)$ for all μ, σ and $P \in [P_{\mathrm{start}}^p, P_{\mathrm{end}}^p]$

Allocate $\mathbf{M}^{\mathrm{occ},p}$ and $\mathbf{M}^{\mathrm{virt},p}$ as copy of $\mathbf{S}^p$

Assign each process a second group q

Create ranges $[\mu_{\mathrm{start}}^q, \mu_{\mathrm{end}}^q]$ and $[\sigma_{\mathrm{start}}^q, \sigma_{\mathrm{end}}^q]$ for primary basis functions

Allocate DBCSR matrices $\mathbf{M}^{\mathrm{occ/virt},q}$ in the q group (elements $M_{(\mu\sigma)Q}^{\mathrm{occ/virt},q}$, combined row index $(\mu\sigma)$, column index Q for all Q and $\mu \in [\mu_{\mathrm{start}}^q, \mu_{\mathrm{end}}^q]$, $\sigma \in [\sigma_{\mathrm{start}}^q, \sigma_{\mathrm{end}}^q]$; single block for $(\mu\sigma)$ index, atom blocks for Q)

Do $\tau_j = \tau_1, \tau_2, \ldots, \tau_M$

$\tilde{C}_{\mu i}^{\mathrm{occ}}(\tau_j) = C_{\mu i} e^{-|(\varepsilon_i - \varepsilon_{\mathrm{F}})\tau_j|/2}$, i: occupied orbital

$\tilde{C}_{\mu a}^{\mathrm{virt}}(\tau_j) = C_{\mu a} e^{-|(\varepsilon_a - \varepsilon_{\mathrm{F}})\tau_j|/2}$, a: virtual orbital

$\mathbf{D}^{\mathrm{occ}}(\tau_j) = [\tilde{\mathbf{C}}^{\mathrm{occ}}(\tau_j)]^{\mathrm{T}} \tilde{\mathbf{C}}^{\mathrm{occ}}(\tau_j)$ (ScaLAPACK)

$\mathbf{D}^{\mathrm{virt}}(\tau_j) = [\tilde{\mathbf{C}}^{\mathrm{virt}}(\tau_j)]^{\mathrm{T}} \tilde{\mathbf{C}}^{\mathrm{virt}}(\tau_j)$ (ScaLAPACK)

Replicate $\mathbf{D}^{\mathrm{occ}}(\tau_j)$ and $\mathbf{D}^{\mathrm{virt}}(\tau_j)$ to every group p and store them in a DBCSR matrix (atom blocks for rows and columns, respectively)

$\mathbf{M}^{\mathrm{occ},p}(\tau_j) = \mathbf{D}^{\mathrm{occ}}(\tau_j)\mathbf{S}^p$ (DBCSR, locally in p group)

$\mathbf{M}^{\mathrm{virt},p}(\tau_j) = \mathbf{D}^{\mathrm{virt}}(\tau_j)\mathbf{S}^p$ (DBCSR, locally in p group)

Reorder data from p groups to q groups: $\mathbf{M}^{\mathrm{occ/virt},p}(\tau_j)$ to $\mathbf{M}^{\mathrm{occ/virt},q}(\tau_j)$

$\mathbf{P}^q(\tau_j) = [\mathbf{M}^{\mathrm{occ},q}(\tau_j)]^{\mathrm{T}} \mathbf{M}^{\mathrm{virt},q}(\tau_j)$ (DBCSR, locally in q group)

Sum up $\mathbf{P}^q(\tau_j)$ from every q group, fill it into $\mathbf{P}(\tau_j) = \sum_q \mathbf{P}^q(\tau_j)$ and spread $\mathbf{P}(\tau_j)$ to all processes

$\mathbf{Q}(\tau_j) = \mathbf{K}^{\mathrm{T}} \mathbf{P}(\tau_j) \mathbf{K}$ (ScaLAPACK)

Do $\omega_k = \omega_1, \omega_2, \ldots, \omega_M$

Fourier $\tau \to \omega$: $\mathbf{Q}(\omega_k) = \mathbf{Q}(\omega_k) + 2\lambda_{kj}\mathbf{Q}(\tau_j)\cos(\tau_j\omega_k)$

End do ω_k

End do τ_j

Set $E_{\mathrm{c}}^{\mathrm{PPA}} = 0$

Do $\omega_k = \omega_1, \omega_2, \ldots, \omega_M$

$E_{\mathrm{c}}^{\mathrm{PPA}} = E_{\mathrm{c}}^{\mathrm{PPA}} + \sigma_k[\ln[\det(\mathbf{1} + \mathbf{Q}(\omega_k))] - \mathrm{Tr}(\mathbf{Q}(\omega_k))]$ (ScaLAPACK)

End do ω_k

Figure 2: Parallel implementation of the algorithm to compute the RPA correlation energy in $\mathcal{O}(N^3)$ operations. Some matrices and their indices have already been defined in Fig. 1. As dense linear algebra library, we employ ScaLAPACK. All calls to ScaLAPACK are executed employing all processes.

The pseudocode for the parallel implementation of the cubic-scaling RPA algorithm is presented in Fig. 2. In the following section, we discuss this figure in detail.

5.1 General strategy for the parallel implementation

For the parallelization, we are guided by three strategies: First, the three-index tensors $(\mu\nu P)$, $M_{\mu\sigma R}^{\mathrm{occ}}(\tau)$ and $M_{\mu\sigma T}^{\mathrm{virt}}(\tau)$ (as defined Fig. 1) are not replicated due to the huge amount of memory needed for these tensors. Second, all two-index matrices as $D_{\mu\lambda}^{\mathrm{occ}}(\tau)$, $D_{\nu\sigma}^{\mathrm{virt}}(\tau)$ and

$P_{RT}(\tau)$ (as defined Fig. 1) are replicated into small subgroups. Third, all sparse matrix-matrix multiplications are carried out in these small subgroups in order to reduce the communication needed for the sparse matrix-matrix multiplications.

5.2 Parallel matrix-tensor operations

We define two different MPI subgroups p and q. Every p group hosts a range $[P^p_{\text{start}}, P^p_{\text{end}}]$ of RI basis functions. After computing the scaled density matrices $D^{\text{occ}}_{\mu\lambda}(\tau)$ and $D^{\text{virt}}_{\nu\sigma}(\tau)$ employing all processes and the dense linear algebra library ScaLAPACK, $D^{\text{occ}}_{\mu\lambda}(\tau)$ and $D^{\text{virt}}_{\nu\sigma}(\tau)$ are replicated to every p group. Then, the multiplication of $D^{\text{occ}}_{\mu\lambda}(\tau)$ and $D^{\text{virt}}_{\nu\sigma}(\tau)$ with the three-center overlap integrals $(\mu\nu P)$ [Eq. (29)] is carried out by DBCSR[57] locally in the p group for $P \in [P^p_{\text{start}}, P^p_{\text{end}}]$ and $\mu, \sigma \in [1, N_{\text{prim}}]$:

$$M^{\text{occ/virt},p}_{\mu\sigma P}(\tau) = \sum_{\lambda=1}^{N_{\text{prim}}} (\lambda\sigma P) D^{\text{occ/virt}}_{\mu\lambda}(\tau)\,, \tag{35}$$

where N_{prim} is the number of primary basis functions.

After completing the computation from Eq. (35), we redistribute $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ from p groups to q groups: The subgroup q is defined by cutting the index pair $\mu\sigma$ of $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ into ranges $[\mu^q_{\text{start}}, \mu^q_{\text{end}}]$ and $[\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ covering all primary basis functions:

$$\bigcup_{q=1}^{N_q} [\mu^q_{\text{start}}, \mu^q_{\text{end}}] \times [\sigma^q_{\text{start}}, \sigma^q_{\text{end}}] = [1, N_{\text{prim}}] \times [1, N_{\text{prim}}]\,, \tag{36}$$

where N_q is the number of q groups. A q group hosts $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ for $\mu \in [\mu^q_{\text{start}}, \mu^q_{\text{end}}]$, $\sigma \in [\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ and all RI basis functions R, T. Each of $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ is stored in the q group in a single DBCSR matrix with a single row block consisting of a combined $(\mu\sigma)$ index and atom blocks for the RI index R and T, respectively. The ranges $[\mu^q_{\text{start}}, \mu^q_{\text{end}}]$ and $[\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ are chosen such that all $\mu \in [\mu^q_{\text{start}}, \mu^q_{\text{end}}]$ and all $\sigma \in [\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ belong to neighboring atoms, respectively. Then, $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ in the q group are sparse in

the combined row index ($\mu\sigma$) and the column index R/T. The DBCSR matrix $P_{RT}(\tau)$ is replicated in the q group and the operation from Eq. (31) is carried out locally in the q group for the given ranges $[\mu^q_{\text{start}}, \mu^q_{\text{end}}]$ and $[\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ and all R, T:

$$P^q_{RT}(\tau) = \sum_{\mu=\mu^q_{\text{start}}}^{\mu^q_{\text{end}}} \sum_{\sigma=\sigma^q_{\text{start}}}^{\sigma^q_{\text{end}}} M^{\text{occ},q}_{\mu\sigma R}(\tau) M^{\text{virt},q}_{\mu\sigma T}(\tau)\,. \tag{37}$$

Then, all q-local DBCSR matrices $P^q_{RT}(\tau)$ are summed up to obtain the full $P_{RT}(\tau)$ matrix:

$$P_{RT}(\tau) = \sum_{q=1}^{N_q} P^q_{RT}(\tau)\,. \tag{38}$$

The p and q groups are chosen such that they are as small as possible to minimize the communication for the sparse matrix-matrix multiplication. On the other hand, p and q groups have to be as large as necessary not to run out of memory since the replicated matrices $D^{\text{occ}}_{\mu\lambda}(\tau)$, $D^{\text{virt}}_{\nu\sigma}(\tau)$ and $P_{RT}(\tau)$ can require a large amount of memory.

The matrix operation $\mathbf{Q}(\tau) = \mathbf{K}^{\text{T}}\mathbf{P}(\tau)\mathbf{K}$ is carried out using full matrices, all processes and ScaLAPACK. We compute the determinant and the trace of $\mathbf{Q}(\omega)$ to arrive at the RPA correlation energy as in Ref. 45, see Fig. 2.

5.3 Memory reduction by an iterative scheme

The memory needed for $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ can be reduced by an additional loop over $\mu\sigma$ blocks (not sketched in Fig. 2): We break the ranges $[\mu^q_{\text{start}}, \mu^q_{\text{end}}]$ and $[\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]$ of every q group again in N_{cut} batches $[\mu^{q,s}_{\text{start}}, \mu^{q,s}_{\text{end}}]$ and $[\sigma^{q,s}_{\text{start}}, \sigma^{q,s}_{\text{end}}]$ with

$$\bigcup_{s=1}^{N_{\text{cut}}} [\mu^{q,s}_{\text{start}}, \mu^{q,s}_{\text{end}}] \times [\sigma^{q,s}_{\text{start}}, \sigma^{q,s}_{\text{end}}] = [\mu^q_{\text{start}}, \mu^q_{\text{end}}] \times [\sigma^q_{\text{start}}, \sigma^q_{\text{end}}]\,. \tag{39}$$

Then, $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ are computed from Eq. (35) for $\mu \in [\mu^{q,s}_{\text{start}}, \mu^{q,s}_{\text{end}}]$ and $\sigma \in [\sigma^{q,s}_{\text{start}}, \sigma^{q,s}_{\text{end}}]$ which reduces the memory of $M^{\text{occ}}_{\mu\sigma R}(\tau)$ and $M^{\text{virt}}_{\mu\sigma T}(\tau)$ by a factor N_{cut}. Subsequently, $M^{\text{occ}}_{\mu\sigma R}(\tau)$

and $M^{\mathrm{virt}}_{\mu\sigma T}(\tau)$ are contracted [Eqs. (31) and (37)],

$$P^{q,s}_{RT}(\tau) = \sum_{\mu=\mu^{q,s}_{\mathrm{start}}}^{\mu^{q,s}_{\mathrm{end}}} \sum_{\sigma=\sigma^{q,s}_{\mathrm{start}}}^{\sigma^{q,s}_{\mathrm{end}}} M^{\mathrm{occ},q,s}_{\mu\sigma R}(\tau) M^{\mathrm{virt},q,s}_{\mu\sigma T}(\tau) \tag{40}$$

and the memory of $M^{\mathrm{occ},q,s}_{\mu\sigma R}(\tau)$ and $M^{\mathrm{virt},q,s}_{\mu\sigma T}(\tau)$ is freed. We obtain $P^{s}_{RT}(\tau)$ for a batch s by summing up from the q groups,

$$P^{s}_{RT}(\tau) = \sum_{q=1}^{N_q} P^{q,s}_{RT}(\tau)\,, \tag{41}$$

and $P_{RT}(\tau)$ is computed by the loop over the s batches:

$$P_{RT}(\tau) = \sum_{s=1}^{N_{\mathrm{cut}}} P^{s}_{RT}(\tau)\,. \tag{42}$$

As already mentioned, we obtain a memory reduction for $M^{\mathrm{occ}}_{\mu\sigma R}(\tau)$ and $M^{\mathrm{virt}}_{\mu\sigma T}(\tau)$ by a factor of N_{cut} by this procedure where no additional computations are necessary.

6 Validation

In this section, we compare the RPA correlation energy computed by the $\mathcal{O}(N^3)$ algorithm to the RPA correlation energy computed by the canonical $\mathcal{O}(N^4)$ algorithm[33,44,45]. As starting point for the RPA correlation energy calculation, we employ DFT with the PBE functional[70].

6.1 Quartic-scaling RPA with the overlap metric vs. cubic-scaling RPA with the overlap metric

We begin with a comparison of cubic-scaling RPA and quartic-scaling RPA employing the overlap metric[54] in both cases, see Table 1. The only difference between both algorithms is the Fourier transform from τ to ω [Eqs. (23) and (33)] which is needed for the cubic-scaling

Table 1: Convergence of the RPA correlation energy of 32 water molecules in a periodic box with the number of grid points M. We use the same primary cc-TZVP basis and RI-cc-TZVP Ri-basis for two different RPA algorithms: The canonical RI-RPA algorithm by Eshuis *et al.*[33] with the overlap metric and the cubic-scaling RPA algorithm proposed in this work.

Number of grid points M	RPA correlation energy in Hartree computed with	
	$\mathcal{O}(N^4)$ RPA (overl. metric)	$\mathcal{O}(N^3)$ RPA (overl. metric)
6	−13.026873930	−13.028861136
8	−13.028957756	−13.028909098
10	−13.028897327	−13.028898786
12	−13.028899938	−13.028899877
14	−13.028899831	−13.028899833
16	−13.028899834	−13.028899834
18	−13.028899834	−13.028899834
20	−13.028899834	−13.028899834

RPA algorithm and absent in the quartic-scaling RPA algorithm. The minimax grid for the frequency integration [Eqs. (21) and (34), respectively] is identical for both cases. As it can be seen from Table 1, both algorithms converge quickly with the number of minimax points: An accuracy of ten digits is already reached for fourteen minimax points for both algorithms. This observation is in agreement with Ref. 48. We conclude that the accuracy of the RPA correlation energy is not affected by the additional Fourier transform from time to frequency.

6.2 Quartic-scaling RPA with the Coulomb metric vs. cubic-scaling RPA with the overlap metric

As a second test, we compare the canonical RPA with the Coulomb metric to the cubic-scaling RPA with the overlap metric, see Table 2. These algorithms share the same primary and RI basis. As shown in Table 2, the cubic-scaling RPA correlation energy deviates by 71 μH from the quartic-scaling RPA correlation energy using the Coulomb metric. In contrast, the cubic-scaling RPA correlation energy agrees within an accuracy better than 10^{-3} μH

with the quartic-scaling RPA correlation energy using the overlap metric, see Table 1. We conclude, that there is a deviation between the RPA results using the overlap metric and the RPA results using the Coulomb metric, but the deviation is small.

Table 2: RPA correlation energy of 32 water molecules computed by three different algorithms. All algorithms share the same primary and RI basis (cc-TZVP and RI-cc-TZVP, respectively). The RI-cc-TZVP basis has been generated as described in Ref. 44 and 71. For all RPA calculations, 20 quadrature points for the minimax grids have been used.

Method	E_c^{RPA} in Hartree
$\mathcal{O}(N^4)$ RPA (im. freq., Coulomb metric)	−13.0250
$\mathcal{O}(N^4)$ RPA (im. freq., overlap metric)	−13.0289
$\mathcal{O}(N^3)$ RPA (im. time, overlap metric)	−13.0289

6.3 Effect of sparsity

Finally, we investigate the effect of filtering blocks of sparse matrices occuring in the cubic-scaling RPA algorithm, see Table 3: We choose the filter coefficients 10^{-8} and $3 \cdot 10^{-6}$ for the atom blocks (for details on these coefficients, see caption of the table) such that the relative accuracy of the RPA correlation energy is 0.01 % compared to the non-filtered result. We observe that the execution time is reduced by a factor three when this filter criterion is applied and we conclude, that already two third of the computations can be avoided for 32 water molecules in a cubic box. For all following scaling benchmarks, we employ the filter criteria 10^{-8} and $3 \cdot 10^{-6}$, respectively.

7 Benchmark calculations on the system size scaling

After validating our implementation, we turn over to investigate the scaling of the execution time with respect to the system size. As in Sec. 6, we use a cc-TZVP primary basis with corresponding RI basis[12,44,45,71], the parameters $\varepsilon_{\text{filter prim}} = 3 \cdot 10^{-6}$, $\varepsilon_{\text{filter RI}} = 10^{-8}$

Table 3: RPA correlation energy of 32 water molecules computed by the cubic-scaling algorithm and two different filtering thresholds for removing blocks from sparse matrices. Every block belongs to basis functions of an atom pair. The filtering threshold for $N_{\mathrm{RI}} \times N_{\mathrm{RI}}$ matrices refers to filtering $\mathbf{P}(\tau)$ in Eq. (31). The filtering threshold for $N_{\mathrm{prim}} \times N_{\mathrm{prim}}$ matrices refers to filtering $\mathbf{D}^{\mathrm{occ}}(\tau)$, $\mathbf{D}^{\mathrm{virt}}(\tau)$ and all three-index tensors. The execution time was measured on 576 CRAY-XC40 cores. With the filtering thresholds of 10^{-8} and $3 \cdot 10^{-6}$ as given in the table, the relative change of the RPA correlation energy due to the filtering is 0.01 % where the computation time is reduced by a factor of three.

Filtering threshold for blocks in a		$E_{\mathrm{c}}^{\mathrm{RPA}}$ in Hartree	Execution time in s
$N_{\mathrm{RI}} \times N_{\mathrm{RI}}$ matrix	$N_{\mathrm{prim}} \times N_{\mathrm{prim}}$ matrix		
10^{-8}	$3 \cdot 10^{-6}$	– 13.02970	260
no filtering	no filtering	– 13.02890	751

and a time and frequency grid of 15 points. For all calculations presented in this section, we employ this setting.

7.1 Quartic- vs. cubic-scaling RPA

In Fig. 3 (a), we compare the execution time of $\mathcal{O}(N^4)$- and $\mathcal{O}(N^3)$-scaling RPA as function of the system size for water molecules in a periodic box. We observe that for 32 water molecules, the $\mathcal{O}(N^3)$-scaling RPA algorithm[44,45] requires ten times the execution time of the $\mathcal{O}(N^4)$-scaling RPA algorithm. The reason is that the cubic-scaling RPA algorithm is operating in the $\mu\nu$ product basis, where μ and ν are Gaussian basis functions while the quartic-scaling RPA is implemented in the ia basis, where i is an occupied MO and a a virtual one: In a cc-TZVP basis, approximately fifteen Gaussian basis functions per occupied MO are employed which means one order of magnitude more floating point operations for cubic-scaling RPA compared to quartic-scaling RPA (without filtering). This explains that the cubic-scaling RPA algorithm is unfavorable for small systems compared to the quartic-scaling one. With increasing system size, the favorable scaling of the $\mathcal{O}(N^3)$-RPA algorithm is appearing and the break-even point for the two algorithms is observed between 128 and 256 water molecules. Note that the break-even point is weakly dependent on the basis set size.

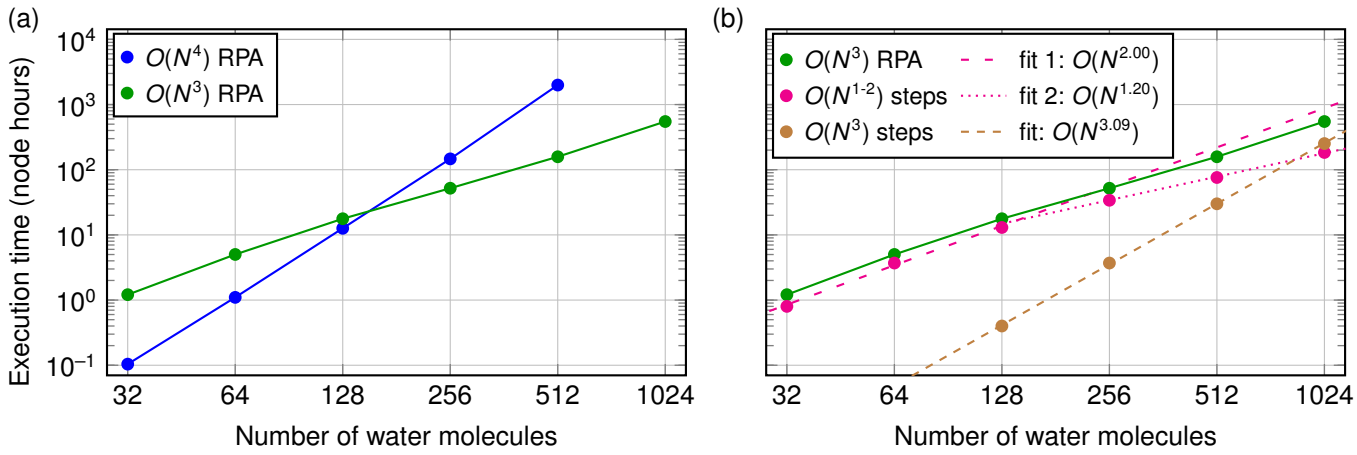

Figure 3: (a) Comparison of the execution time of quartic-scaling RPA (blue dots) and cubic-scaling RPA (green dots) on a CRAY XC40 machine with 36 cores per node. Both methods share the high-quality cc-TZVP basis. The basic cell contains 32 water molecules with a density of $1\,\mathrm{g}/\ell$. The larger systems consist of a $n \times 1 \times 1$ supercell of the 32-water box with $n = 2, 4, 8, 16, 32$. For the largest system with 1024 water molecules (8192 electrons), 58366 primary basis functions and 139264 RI basis functions are used. For small systems, the canonical $\mathcal{O}(N^4)$ RPA method is one order of magnitude faster than the cubic-scaling RPA. The break-even point of both methods lies between 128 and 256 water molecules. For large systems, the cubic-scaling RPA exceeds the canonical RPA in terms of the execution time. (b) Execution time and scaling of intermediate steps of the cubic-scaling RPA algorithm where the total execution time (green marks) is identical to (a). The cubic-scaling steps (brown color) and the linear- and quadratic-scaling steps (magenta) are categorized according to Fig. 1 and Eqs. (29) and (31). The quadratic-scaling steps dominate for small systems. Moreover, the quadratic-scaling steps turn into nearly-linear scaling steps for systems containing 256 water molecules or more since the density matrix is becoming sparse, see Fig. 4. The cubic-scaling steps (brown color) exhibit a small prefactor and are dominating for systems with more than 1000 water molecules.

For 512 water molecules, the cubic-scaling RPA outperforms the canonical RPA in terms of the execution time by one order of magnitude.

7.2 Analyzing the dominant steps in cubic-scaling RPA

In Fig. 3 (b), we break the total execution time of the cubic-scaling RPA algorithm down into cubic-scaling steps and sub-cubic-scaling steps. The cubic-scaling steps (brown color) and the linear- and quadratic-scaling steps (magenta) are categorized according to Fig. 1. We observe that the quadratic-scaling steps dominate for small systems which means that for small systems, the algorithm is effectively quadratic-scaling. Moreover, the quadratic-scaling steps turn into nearly-linear scaling steps for systems containing 256 water molecules or more since the density matrix is becoming sparse, see Fig. 4. The cubic-scaling steps exhibit a small prefactor and are dominating for systems with more than 1000 water molecules. We conclude that the bottleneck for large-scale RPA calculations is the matrix-matrix multiplication from Eq. (32) and the Cholesky decomposition of $\mathbf{1} + \mathbf{Q}(\omega)$ to compute the determinant in Eq. (34).

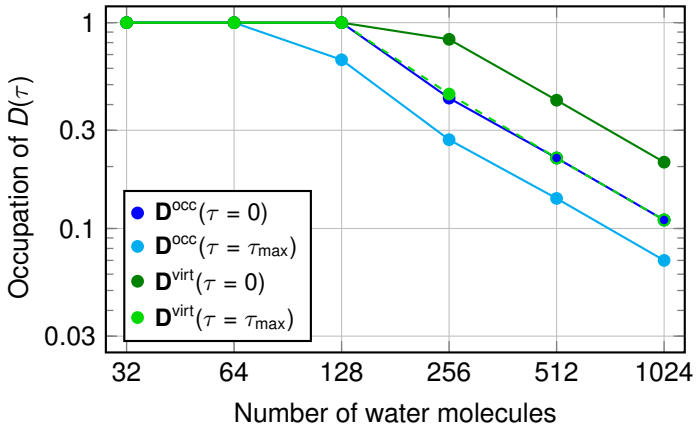

Figure 4: Occupation of the scaled density matrices from Eq. (27) with non-zero elements for the systems from Fig. 3. The matrix blocks correspond to atoms and are filtered with a threshold of 10^{-6} for the Frobenius norm of the block.[57] A similar occupation of scaled density matrices has been reported in Ref. 67.

7.3 Truly three-dimensional calculations

In Fig. 3, the basic cell of 32 water molecules was repeated in one dimension, e. g. an $n\times1\times1$ supercell was employed. For low-scaling algorithms in a Gaussian basis, low-dimensional systems with large extend in one dimension are favorable since there are many pairs of Gaussian basis functions which are far away from each other and therefore have zero overlap. To test how the cubic-scaling RPA algorithm performs for truly three-dimensional systems, we show the execution time for the cubic water supercells $2\times2\times2$ and $3\times3\times3$ in Fig. 5: We observe a scaling $\mathcal{O}(N^{2.16})$ for the execution time from $1\times1\times1$ to $3\times3\times3$. As it can be seen from Fig. 5, the execution time of the cubic $n\times n\times n$ supercell slightly exceeds the execution time for the corresponding $n^3\times1\times1$ supercell. The reason is that the scaled density matrices are not yet sparse for the $n\times n\times n$ supercell ($n=2,3$) in contrast to the linear chain of supercells, see Fig. 4. We conclude that the cubic-scaling RPA algorithm from this work also scales well for systems which are extended in all three dimensions.

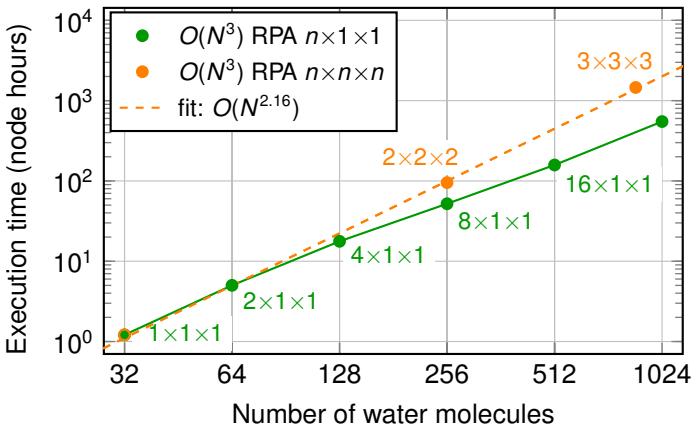

Figure 5: Execution time of cubic-scaling RPA for $n\times1\times1$ supercells (green, as in Fig. 3) and $n\times n\times n$ supercells (orange) on a CRAY XC40 machine with 36 cores per node. The single cell consists of 32 water molecules. The execution time of the cubic supercells (orange) is exceeding the one of the linear supercells (green) since the density matrix of linear cells is getting sparse, see Fig. 4, in contrast to the density matrix of cubic cells.

8 Conclusion and Outlook

We have presented an algorithm for computing the correlation energy in the random phase approximation (RPA) in a Gaussian basis requiring $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ memory. The method is a combination of several known techniques: As first technique, we employ the resolution of the identity (RI) with the overlap metric as suggested by various authors[53–56]. The advantage of RI with the overlap metric compared to RI with the Coulomb metric is that the occurring three-center overlap integrals are sparse[55,56]. To exploit the sparsity of the three-center overlap integrals, we reformulate RI-RPA in the Gaussian basis as suggested by Schurkus and Ochsenfeld[53]. Since the density response has a convenient analytic form in imaginary time, we follow Rojas *et al.*[64] to compute the density response in the RI basis in imaginary time. For the Fourier transform from imaginary time to imaginary frequency, we employ the minimax grids as proposed in the seminal work of Kaltak *et al.*[48] The final formula for computing the RPA correlation energy by an integral over imaginary frequencies was established by Eshuis *et al.*[33] No localization schemes[50] or stochastic approximations[51,52] are needed in our algorithm.

Since the computation of the RPA correlation energy is still of high computational cost, an efficient parallelization is crucial, especially for the sparse matrix operations. For this purpose, we use the sparse linear algebra library DBCSR[57] which turns out to be highly efficient. Due to the low-scaling characteristics of the algorithm together with its efficient parallel implementation, we could apply cubic-scaling RPA up to a thousand water molecules using a cc-TZVP basis.

Our work can be seen as a prototype for a low-scaling wavefunction-based method using a global RI with the overlap metric. With the techniques presented in this work, efficient low-scaling algorithms can be designed for Hartree-Fock[72–79], Laplace-SOS-MP2[80–86] and *GW*[48,87–90].

Acknowledgement

Calculations were enabled by the Swiss National Supercomputing Center (CSCS) under project ID mr2 and uzh1. We thank Andreas Glöß and Joost VandeVondele for helpful discussions. This research was supported by the NCCR MARVEL, funded by the Swiss National Science Foundation.

Supporting Information Available

The derivation of Eq. (21) from Eq. (16) and an exemplary input file of CP2K is available as supporting information. This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Eshuis, H.; Bates, J.; Furche, F. *Theor. Chem. Acc.* **2012**, *131*, 1–18.

(2) Ren, X.; Rinke, P.; Joas, C.; Scheffler, M. *J. Mater. Sci.* **2012**, *47*, 7447–7471.

(3) Johansson, M. P.; Warnke, I.; Le, A.; Furche, F. *J. Phys. Chem. C* **2014**, *118*, 29370–29377.

(4) Eshuis, H.; Furche, F. *J. Phys. Chem. Lett.* **2011**, *2*, 983–989.

(5) Fuchs, M.; Gonze, X. *Phys. Rev. B* **2002**, *65*, 235109.

(6) Furche, F.; Van Voorhis, T. *J. Chem. Phys.* **2005**, *122*, 164106.

(7) Heßelmann, A.; Görling, A. *Mol. Phys.* **2011**, *109*, 2473.

(8) Toulouse, J.; Gerber, I. C.; Jansen, G.; Savin, A.; Ángyán, J. G. *Phys. Rev. Lett.* **2009**, *102*, 096404.

(9) Furche, F. *Phys. Rev. B* **2001**, *64*, 195120.

(10) Aryasetiawan, F.; Miyake, T.; Terakura, K. *Phys. Rev. Lett.* **2002**, *88*, 166401.

(11) Nguyen, H.-V.; de Gironcoli, S. *Phys. Rev. B* **2009**, *79*, 205114.

(12) Del Ben, M.; VandeVondele, J.; Slater, B. *J. Phys. Chem. Lett.* **2014**, *5*, 4122–4128.

(13) Spreafico, C.; VandeVondele, J. *Phys. Chem. Chem. Phys.* **2014**, *16*, 26144–26152.

(14) Spreafico, C.; VandeVondele, J. *J. Phys. Chem. C* **2015**, *119*, 15009–15018.

(15) Del Ben, M.; Hutter, J.; VandeVondele, J. *J. Chem. Phys.* **2015**, *143*, 054506.

(16) Schimka, L.; Gaudoin, R.; Klimeš, J.; Marsman, M.; Kresse, G. *Phys. Rev. B* **2013**, *87*, 214102.

(17) Lebègue, S.; Harl, J.; Gould, T.; Ángyán, J.; Kresse, G.; Dobson, J. *Phys. Rev. Lett* **2010**, *105*, 196401.

(18) Ren, X.; Rinke, P.; Scheffler, M. *Phys. Rev. B* **2009**, *80*, 045402.

(19) Olsen, T.; Yan, J.; Mortensen, J. J.; Thygesen, K. S. *Phys. Rev. Lett.* **2011**, *107*, 156401.

(20) Mittendorfer, F.; Garhofer, A.; Redinger, J.; Klimeš, J.; Harl, J.; Kresse, G. *Phys. Rev. B* **2011**, *84*, 201401.

(21) Schimka, L.; Harl, J.; Stroppa, A.; Grüneis, A.; Marsman, M.; Mittendorfer, F.; Kresse, G. *Nat. Mater.* **2010**, *9*, 741–744.

(22) Harl, J.; Schimka, L.; Kresse, G. *Phys. Rev. B* **2010**, *81*, 115126.

(23) Harl, J.; Kresse, G. *Phys. Rev. B* **2008**, *77*, 045136.

(24) Macher, M.; Klimeš, J.; Franchini, C.; Kresse, G. *J. Chem. Phys.* **2014**, *140*, 084502.

(25) Cheng, J.; VandeVondele, J. *Phys. Rev. Lett.* **2016**, *116*, 086402.

(26) Miyake, T.; Aryasetiawan, F.; Kotani, T.; van Schilfgaarde, M.; Usuda, M.; Terakura, K. *Phys. Rev. B* **2002**, *66*, 245103.

(27) García-González, P.; Fernández, J. J.; Marini, A.; Rubio, A. *J. Phys. Chem. A* **2007**, *111*, 12458–12465.

(28) Xiao, B.; Sun, J.; Ruzsinszky, A.; Feng, J.; Perdew, J. P. *Phys. Rev. B* **2012**, *86*, 094109.

(29) Rohlfing, M.; Bredow, T. *Phys. Rev. Lett.* **2008**, *101*, 266106.

(30) Marini, A.; García-González, P.; Rubio, A. *Phys. Rev. Lett.* **2006**, *96*, 136404.

(31) Lu, D.; Li, Y.; Rocca, D.; Galli, G. *Phys. Rev. Lett.* **2009**, *102*, 206411.

(32) Li, Y.; Lu, D.; Nguyen, H.-V.; Galli, G. *J. Phys. Chem. A* **2010**, *114*, 1944–1952.

(33) Eshuis, H.; Yarkony, J.; Furche, F. *J. Chem. Phys.* **2010**, *132*, 234114.

(34) Göltl, F.; Grüneis, A.; Bučko, T.; Hafner, J. *J. Chem. Phys.* **2012**, *137*, 114111.

(35) Paier, J.; Ren, X.; Rinke, P.; Scuseria, G. E.; Grüneis, A.; Kresse, G.; Scheffler, M. *New J. Phys.* **2012**, *14*, 043002.

(36) Grimme, S.; Steinmetz, M. *Phys. Chem. Chem. Phys.* **2016**, *18*, 20926–20937.

(37) Toulouse, J.; Zhu, W.; Ángyán, J. G.; Savin, A. *Phys. Rev. A* **2010**, *82*, 032502.

(38) Zhu, W.; Toulouse, J.; Savin, A.; Ángyán, J. G. *J. Chem. Phys.* **2010**, *132*, 244108.

(39) Toulouse, J.; Gerber, I. C.; Jansen, G.; Savin, A.; Ángyán, J. G. *Phys. Rev. Lett.* **2009**, *102*, 096404.

(40) Del Ben, M.; Schönherr, M.; Hutter, J.; VandeVondele, J. *J. Phys. Chem. Lett.* **2013**, *4*, 3753–3759.

(41) Harl, J.; Kresse, G. *Phys. Rev. Lett.* **2009**, *103*, 056401.

(42) Olsen, T.; Thygesen, K. S. *Phys. Rev. B* **2013**, *87*, 075111.

(43) Rocca, D. *J. Chem. Phys.* **2014**, *140*, 18A501.

(44) Del Ben, M.; Hutter, J.; VandeVondele, J. *J. Chem. Theory Comput.* **2013**, *9*, 2654–2671.

(45) Del Ben, M.; Schütt, O.; Wentz, T.; Messmer, P.; Hutter, J.; VandeVondele, J. *Comput. Phys. Commun.* **2015**, *187*, 120–129.

(46) Ren, X.; Rinke, P.; Blum, V.; Wieferink, J.; Tkatchenko, A.; Sanfilippo, A.; Reuter, K.; Scheffler, M. *New J. Phys.* **2012**, *14*, 053020.

(47) Moussa, J. E. *J. Chem. Phys.* **2014**, *140*, 014107.

(48) Kaltak, M.; Klimeš, J.; Kresse, G. *J. Chem. Theory Comput.* **2014**, *10*, 2498–2507.

(49) Kaltak, M.; Klimeš, J.; Kresse, G. *Phys. Rev. B* **2014**, *90*, 054115.

(50) Kállay, M. *J. Chem. Phys.* **2015**, *142*, 204105.

(51) Neuhauser, D.; Rabani, E.; Baer, R. *J. Phys. Chem. Lett.* **2013**, *4*, 1172–1176.

(52) Gao, Y.; Neuhauser, D.; Baer, R.; Rabani, E. *J. Chem. Phys.* **2015**, *142*, 034106.

(53) Schurkus, H. F.; Ochsenfeld, C. *J. Chem. Phys.* **2016**, *144*, 031101.

(54) Vahtras, O.; Almlöf, J.; Feyereisen, M. *Chem. Phys. Lett.* **1993**, *213*, 514.

(55) Jung, Y.; Sodt, A.; Gill, P. M.; Head-Gordon, M. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6692–6697.

(56) Reine, S.; Tellgren, E.; Krapp, A.; Kjærgaard, T.; Helgaker, T.; Jansik, B.; Høst, S.; Salek, P. *J. Chem. Phys.* **2008**, *129*.

(57) Borštnik, U.; VandeVondele, J.; Weber, V.; Hutter, J. *Parallel Comput.* **2014**, *40*, 47–58.

(58) Schütt, O.; Messmer, P.; Hutter, J.; VandeVondele, J. *Electronic Structure Calculations on Graphics Processing Units*; John Wiley & Sons, Ltd, 2016; pp 173–190.

(59) Obara, S.; Saika, A. *J. Chem. Phys.* **1986**, *84*, 3963.

(60) Reine, S.; Tellgren, E.; Helgaker, T. *Phys. Chem. Chem. Phys.* **2007**, *9*, 4771–4779.

(61) Braess, D.; Hackbusch, W. *SIAM J. Numer. Anal.* **2005**, *25*, 685–697.

(62) Ewald, P. P. *Ann. Phys.* **1921**, *369*, 253–287.

(63) Lengreth, D. C.; Perdew, J. P. *Phys. Rev. B* **1977**, *15*, 2884.

(64) Rojas, H. N.; Godby, R. W.; Needs, R. J. *Phys. Rev. Lett.* **1995**, *74*, 1827.

(65) Häser, M.; Almlöf, J. *J. Chem. Phys.* **1992**, *96*, 489.

(66) Doser, B.; Lambrecht, D. S.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2009**, *130*, 064107.

(67) Maurer, S. A.; Clin, L.; Ochsenfeld, C. *J. Chem. Phys.* **2014**, *140*, 224112.

(68) Takatsuka, A.; Ten-no, S.; Hackbusch, W. *J. Chem. Phys.* **2008**, *129*, 044112.

(69) Kats, D.; Usvyat, D.; Loibl, S.; Merz, T.; Schütz, M. *J. Chem. Phys.* **2009**, *130*, 127101.

(70) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.

(71) Weigend, F.; Köhn, A.; Hättig, C. *J. Chem. Phys.* **2002**, *116*, 3175–3183.

(72) Ihrig, A. C.; Wieferink, J.; Zhang, I. Y.; Ropo, M.; Ren, X.; Rinke, P.; Scheffler, M.; Blum, V. *New J. Phys.* **2015**, *17*, 093020.

(73) Levchenko, S. V.; Ren, X.; Wieferink, J.; Johanni, R.; Rinke, P.; Blum, V.; Scheffler, M. *Comp. Phys. Comm.* **2015**, *192*, 60–69.

(74) Sodt, A.; Head-Gordon, M. *J. Chem. Phys.* **2008**, *128*.

(75) Della Sala, F.; Görling, A. *J. Chem. Phys.* **2001**, *115*.

(76) Köppl, C.; Werner, H.-J. *J. Chem. Theory Comput.* **2016**, *12*, 3122–3134.

(77) Rebolini, E.; Izsák, R.; Reine, S. S.; Helgaker, T.; Pedersen, T. B. *J. Chem. Theory Comput.* **2016**, *12*, 3514–3522.

(78) Merlot, P.; Kjærgaard, T.; Helgaker, T.; Lindh, R.; Aquilante, F.; Reine, S.; Pedersen, T. B. *J. Comput. Chem.* **2013**, *34*, 1486–1496.

(79) Dziedzic, J.; Hill, Q.; Skylaris, C.-K. *J. Chem. Phys.* **2013**, *139*.

(80) Pisani, C.; Maschio, L.; Casassa, S.; Halo, M.; Schütz, M.; Usvyat, D. *J. Comput. Chem.* **2008**, *29*, 2113–2124.

(81) Maschio, L. *J. Chem. Theory Comput.* **2011**, *7*, 2818–2830.

(82) Maurer, S. A.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2014**, *141*, 051106.

(83) Kats, D.; Usvyat, D.; Schütz, M. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3430–3439.

(84) Izmaylov, A. F.; Scuseria, G. E. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3421–3429.

(85) Ayala, P. Y.; Kudin, K. N.; Scuseria, G. E. *J. Chem. Phys.* **2001**, *115*, 9698–9707.

(86) Ayala, P. Y.; Scuseria, G. E. *J. Chem. Phys.* **1999**, *110*, 3660–3671.

(87) Liu, P.; Kaltak, M.; Klimeš, J.; Kresse, G. *Phys. Rev. B* **2016**, *94*, 165109.

(88) Wilhelm, J.; Del Ben, M.; Hutter, J. *J. Chem. Theory Comput.* **2016**, *12*, 3623–3635.

(89) Foerster, D.; Koval, P.; Sánchez-Portal, D. *J. Chem. Phys.* **2011**, *135*, 074105.

(90) Neuhauser, D.; Gao, Y.; Arntsen, C.; Karshenas, C.; Rabani, E.; Baer, R. *Phys. Rev. Lett.* **2014**, *113*, 076402.

Graphical TOC Entry